PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 10/08

Engineering Multi-Agent Systems for Design and Implementation of Ubiquitous Computing

Milene Serrano

Maurício Serrano

Carlos José Pereira de Lucena

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900

RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 10/08
Editor: Prof. Carlos José Pereira de Lucena

ISSN: 0103-9741
Março 2008

Engineering Multi-Agent Systems for the Design and Implementation of Ubiquitous Computing*

Milene Serrano[1], Maurício Serrano[1]
Carlos José Pereira de Lucena[1]

[1] Departamento de Informática – PUC-Rio
Rua Marques de São Vicente 225, 4º andar RDC.
Rio de Janeiro – RJ – Brasil

{milene, mauserr}@les.inf.puc-rio.br
{lucena}@inf.puc-rio.br

Abstract. Agent properties are very relevant for the domain of ubiquitous application. This is especially true for the properties of mobility and autonomy. Moreover, agent collaboration and technologies such as reasoning, learning and goal-oriented planning help to solve one of the main Ubiquitous Computing concerns: to provide services to clients that are in a distributed world and that have only limited mobile devices to access them.

In addition, the Belief Desire Intention (BDI) Model with goals, plans, beliefs, plans deliberations, events and other JADEX agent capabilities is useful to allow for goal-oriented modeling. This modeling approach makes it is easy to maintain the ubiquitous requirement of traceability, including functional and non-functional requirements. Based on this, evolutionary maintenance can be achieved using the previous deployment decisions through the analysis of the goal-oriented model.

Some of challenges for Software Engineering in the realm of ubiquity include dealing with content adaptation, security, mobility, requirement modeling, ethnomethodology (e.g. personalities, capabilities and users' preferences) and different profiles (e.g. network, user, device and content profiles). In this challenging and almost untapped context, we have conducted novel experimental research that investigates the engineering of Multi-Agent Systems in association with Ubiquitous Computing.

Keywords: Software Engineering, Multi-Agent Systems, Ubiquitous Computing, Goal-Oriented Planning, Belief Desire Intention (*BDI*) Model, Wireless Technology, Evolutionary Maintenance.

Resumo. Propriedades dos agentes são muito relevantes para o domínio das aplicações ubíquas. Esse fato é particularmente verdade para as propriedades de autonomia e mobilidade. Além disso, a colaboração de agentes e tecnologias tais como raciocínio, aprendizado e planejamento orientado à meta ajudam a resolver um dos principais problemas da Computação Ubíqua: prover serviços aos clientes que estão em um mundo distribuído e que têm somente dispositivos móveis limitados para acessá-los.

Adicionalmente, o Modelo *BDI* (*Belief Desire Intention*) com metas, planos, crenças, deliberação de planos, eventos e outras capacidades dos agentes *JADEX* permite a modelagem orientada à meta. Essa abordagem de modelagem torna fácil manter a rastreabilidade dos requisitos ubíquos, incluindo requisitos funcionais e não-funcionais. Com base nesses aspectos, a manutenção evolutiva pode ser alcançada usando decisões anteriores de desenvolvimento através da análise do modelo orientado à meta.

Alguns dos desafios da Engenharia de Software na área da Computação Ubíqua incluem lidar com adaptação de conteúdo, segurança, mobilidade, modelagem de requisitos, etnometodologia (ex. preferências, capacidades e personalidades dos usuários) e diferentes perfis (ex. perfis de conteúdo, dispositivo, usuário e rede) . Nesse contexto desafiador e ainda pouco explorado, foi conduzido um trabalho de pesquisa experimental inovador que investiga a engenharia de Sistemas Multi-Agentes em associação com a Computação Ubíqua.

Palavras-chave: Engenharia de Software, Sistemas Multi-Agentes, Computação Ubíqua, Planejamento Orientado à Meta, Modelo *BDI* (*Belief Desire Intention*), Tecnologia Sem-fio e Manutenção Evolutiva.

* This work has been sponsored by the Ministério de Ciência e Tecnologia da Presidência da República Federativa do Brasil. CAPES

In charge of publications

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br
Web site: http://bib-di.inf.puc-rio.br/techreports/

Table of Contents

1 Introduction

In the last few years, there has been an increase in the number of Internet users accessing it through different devices (e.g. mobile phones, *PDAs*, notebooks, desktops and *tabletPCs*) connected via several access network technologies. In this scenario, the need to deal with a new challenging application of Software Engineering, such as Ubiquitous Computing, became a necessity.

This area brings about new and complex challenges for Computer Science. It assumes a world in which people are surrounded by mobile or fixed devices in a computing environment that supports them in almost everything they do. The scientific community needs to pay special attention to some peculiarities of ubiquitous applications in order for people to coexist in perfect harmony with this technological environment. The intelligence of this environment is provided by advanced network technologies and specific interfaces. To cope with these aspects, new software engineering approaches are in demand. For instance, these approaches need to re-structure and systematize many classical Software Engineering methods and practices to also take into consideration the use of several traditional Artificial Intelligence methods and tools.

Unfortunately, the scientific community has reached little consensus about the proper technologies, methodologies and concepts to be used in connection with Ubiquitous Computing. Much of this is because Ubiquitous Computing is a new area and, despite a lot of ongoing research, few results in the literature propose a differentiated and appropriated approach to deal with ubiquitous applications. There is an absence of specific methods and tools to develop this type of application and guidelines are needed to drive this kind of development. Content adaptation, security, mobility, requirement modeling, ethno-methodology (e.g. personalities, capabilities and users' preferences) and different profiles (e.g. network, user, device and content profiles) are some of challenges for Software Engineering in the ubiquitous context. Ubiquitous Computing opens up innovative opportunities for studying collaboration, adaptability, security, mobility and communication, striving for personalization of the ubiquitous intelligent environment.

In this challenging and almost untapped problem domain, we have conducted novel experimental research that investigates several of the uses of Multi-Agent Systems in association with Ubiquitous Computing. Agent properties, such as mobility, adaptation and autonomy are shown to be very useful for obtaining intelligent environments according to our research results.

This document is organized in sections. Section 2 states our main research challenges. Section 3 reviews some related work. Section 4 deals with some central Ubiquitous Computing concerns. Section 5 reports the case study. Finally, section 6 draws some conclusions from the experience acquired in the experimental research and points to some future work.

2 Some Research Challenges

This section considers some of the challenges presented in the introduction of this paper and states upfront our central research hypotheses to deal with them. They are as follows:

The Multi-Agent System design and development paradigm is appropriated to be used in ubiquitous contexts.

Multi-Agent System technologies and concepts are appropriated to investigate new ways to implement ubiquitous applications. Some useful technologies are: (i) the *JADE* (***J**ava **A**gent **DE**velopment Framework*) [Bellifemine et al. 2007] [Chen et al. 2004], *JADE-LEAP* (***JADE L**ightweight **E**xtensible **A**gent **P**latform)* [Caire 2003] and *JADEX* (***JADE EX**tension Framework*) [Pokahr and Braubach 2007 [a]] [Pokahr and Braubach 2007 [b]] [Pokahr and Braubach 2007 [c]] platforms; (ii) agent properties (e.g. mobility, autonomy and adaptation); (iii) agent reasoning and learning; and (iv) goal-oriented planning involving goals, sub-goals, plans, beliefs, capabilities, events and knowledge base.

Mobile Agents allow for dealing with different devices, such as mobile phones, *PDAs* and *tabletPCs*. Several researchers have introduced software mobility as a technology that enables ubiquitous computers to support various services, for which they may not have been initially designed. Some of these works are reported in [Lobato et al. 2006], [Carrillo-Ramos et al. 2005] and [Satoh 2005].

Multi-Agent Systems are used to solve problems in areas that influence Ubiquitous Computing or are influenced by it. Thus, it is reasonable to assume that Multi-Agent Systems may solve problems in Ubiquitous Computing (a transitivity hypothesis).

Goal Orientation and Multi-Agent Systems using a Belief Desire Intention (*BDI*) Model [Braubach et al. 2003] are appropriate technologies to model ubiquitous applications.

Mobile and wireless technologies are the new cutting edge of the modern world. An increasing number of researchers have embraced this new world and are trying to find ways of analyzing it using a Software Engineering perspective. Examples of this kind of work can be found in [Helin et al. 2006] and [Bellavista et al. 2001].

3 Some Related Work

Several research efforts have addressed the general issue of middleware solutions to deal with Ubiquitous Computing challenges and concerns.

The Gaia Project [Moraïtis et al. 2003] [Román et al. 2002], developed by the Illinois University at Urbana-Champaign, is a distributed middleware operating system infrastructure that manages the resources contained in an Intelligent Space and provides support for Ubiquitous Computing. The Cerberus core service of Gaia integrates identification, authentication and context awareness. Gaia allows applications to be partitioned between different computers and to be moved from computer to computer to satisfy the user.

In Brazil, the Embedded Systems and Pervasive Computing Lab of the Federal University of Campina Grande [Embedded Systems and Pervasive Computing Lab Group 2007] [Nokia Group 2005] is a research center, which studies intelligent environments. For example, their Wireless Project focuses on the research of wireless technologies in industrial environments and the emphasis is on the tools and applications deployment to support mobile automation solutions.

The Computer Human Adapted Interaction Research (*CHAI*) Group [CHAI Group 2007] is investigating pervasive computing interfaces such as tabletop interaction and appliances that are embedded in a pervasive environment. The *CHAI* Group developed, for example, a scrutable model framework, called *PersonisAD*, for context-aware services [Assad et al. 2007].

The *NICTA* Group [NICTA Group 2007] [Liu 2006] is searching and implementing solutions to the multitude of connectivity options that will be available to users of future mobile networking technologies. Their challenges are: (i) context analysis, (ii) connection management and (iii) application awareness.

The Mobile Ubiquitous Services & Technologies Group [Mobile Ubiquitous Services & Technologies Group 2007] is studying mobile ubiquitous services and technologies by conducting experiments. Their focus areas are: (i) collecting essential health information using biosensors; (ii) prototype mobile applications that process and track data; and (iii) middleware for ubiquitous environments. The medical domain is the main context for their research [Leijdekkers and Gay 2006] [Gay and Leijdekkers 2006].

Microsoft's EasyLiving Project [Brumitt et al. 2000] is developing a middleware, called InConcert, to aggregate input/output devices dynamically even when they belong to different computer machines. The devices are, for example, keyboards and mice.

A Project, called *BEACH* [Tandler 2001], developed by *GMD-IPSI* and Darmstadt University, supports cooperative and collaborative applications construction using different and distributed devices. It is based in synchronously sharing information between multiple users.

The Aura Project [Garlan et al. 2002] consists of providing an infrastructure for migrating applications from computer to computer as users move about. The focus is the contextual services and not the multiple computer integration to provide the services. Other related work is described in [Kapitza et al. 2006] and [Wegdam et al. 2004].

The research reported in this paper is in part motivated by the fact that there are few known specific approaches for the Ubiquitous Computing from the perspective of Software Engineering. Our work allowed us to investigate the Ubiquitous Computing State-Of-The-Art and we are now able to use the environment to be presented in section 5, as an experimental test bed to evaluate different technologies and to develop an adequate way to approach Ubiquitous Computing using the Multi-Agent System paradigm from the Software Engineering point of view.

4 Central Issues in Ubiquitous Computing

There seems to be general agreement about the central issues in Ubiquitous Computing. The main concerns related to this technology include the distributed computer systems that support different types of access devices, mobile devices limitations, content adjustment and the handling of different profiles.

4.1 Distributed Environment

Ubiquitous Computing is increasingly dependent on distributed computer systems that support different types of access devices (e.g. mobile phones, smartphones, pagers, *PDAs*, desktops, notebooks and others). The main objective of Ubiquitous Computing is to provide services to users anywhere and at any time, as they are part of an intelligent environment that identifies their presences and seeks their full satisfaction.

This intelligent environment involves the world (e.g. every country, state, city, street, home, room and so on). In this context, ubiquitous applications are more complex than pervasive applications, because we have to address the user mobility. We can consider that Ubiquitous Computing aims at the combined Pervasive and Mobile Computing goals. Thus, it also inherits the problems inherent to the two areas.

4.2 Mobile Device Limitations and Content Adjustment

The widely varying characteristics of mobile devices are challenges that Ubiquitous Computing faces to ensure full user satisfaction. Just to illustrate this point, we note that: different client devices support different features and different screen sizes may demand different sized images. Consequently, it is very common when delivering content to mobile devices to vary the image format, size, color depths and other kinds of content adaptations. Some mobile devices limitations are: (i) screen resolution; (ii) screen color depth; (iii) internal memory capacity; (iv) operating system; and (v) content format and size.

4.3 Different Profiles

To apply content adaptations it is necessary to consider different profiles that contain, for example, the peculiarity and characteristics of users, devices, contents and network. We describe each of them below.

- **User Profile**: has user personal information and adaptation preferences that are used to assure that the content is provided in accordance with the user expectations.
- **Device Profile**: has device hardware capabilities (e.g. screen resolution and internal memory capacity) and device software capabilities (e.g. operating system). This profile is very important to the adaptation process, which considers different devices limitations.
- **Content Profile**: has content information such as format, size, resolution, security, origin and others. It is very important to analyze this profile before sending the content to the user device to avoid the user's dissatisfaction.
- **Network Profile**: has network information such as bandwidth, turnaround time and network traffic. This profile is used to analyze the network status and to optimize the network resources use (e.g. redirecting packets between servers to avoid overload).

5 Case Study

We have based our work on the systematic application of methods and tools from the new area of Software Engineering of Multi-Agent Systems [Bellifemine et al. 2007] [Caire 2003] [Pokahr and Braubach 2007 [a]] [Pokahr and Braubach 2007 [c]] [Braubach et al. 2003]. In our work such methods and tools include: (i) Multi-Agent System and Goal-Oriented design approach; (ii) Belief Desire Intention (*BDI*) Model [Braubach et al. 2003]; (iii) Reasoning [Bigus and Bigus 2001] [Agotnes et al. 2007] [Pokahr et al. 2005], Learning [Bigus and Bigus 2001] [Lagos et al. 2007] and Fuzzy Logic [Bigus and Bigus 2001] [Ruano 2005] [Lemaître et al. 2004] to provide analysis based on knowledge bases to offer the best service to the user, and (iv) the integration of the platforms:

JADEX [Pokahr and Braubach 2007 [a]] [Pokahr and Braubach 2007 [b]] [Pokahr and Braubach 2007 [c]] and *JADE-LEAP* [Caire 2003].

The presentation structure of the case study described in this paper, consists of (i) the use of the *JADE-LEAP* platform as an integrated environment, which is provided by using the platform resources such as tools, architecture model and specific execution modes; (ii) the *JADEX* platform to provide an intelligent environment using *BDI* Model according to intentionality, reasoning and other capabilities of the agents; and (iii) the viewpoints in high or low abstraction levels. The presentation structure details are illustrated in Figure 1.

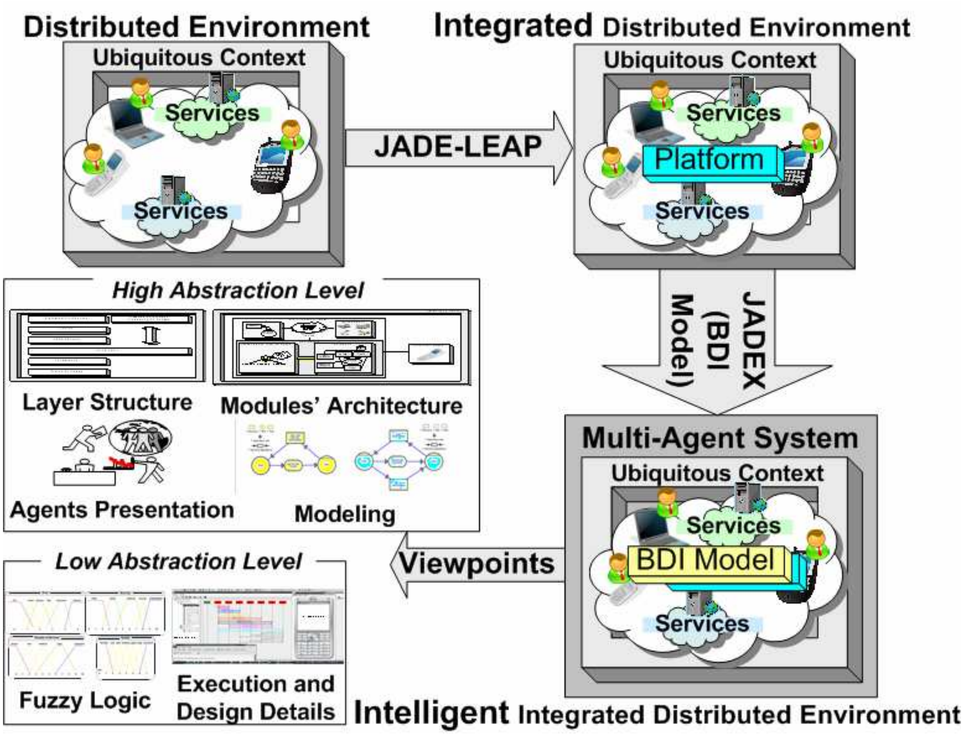

Figure 1: Case Study Presentation

Considering a distributed environment with different mobile devices, users and service remote servers, our challenge was to deal with this ubiquitous context. Thus, the *JADE-LEAP* supports the Ubiquitous Computing concern that considers the inherent distributed computer systems with different types of access devices. This platform has two sub-platforms: the *JADE-LEAP/PJAVA,* which supports powerful devices that run Personal Java; and the *JADE-LEAP/MIDP*1, which supports mobile devices with limited capabilities. Both are described in Section 5.1.

To provide the intelligent environment, presented in section 5.2, we have used *JADEX* and the *BDI* Model. *JADEX* seeks to provide reasoning and learning in Multi-Agent Systems based on goals, plans, beliefs, events and other agent capabilities.

Finally, we have the project viewpoints at the high abstraction level (e.g. Layer Structure, Modules' Architecture, Agents Presentation and Modeling) and at the low abstraction level (e.g. Reasoning with Fuzzy Logic and Application Execution). As follows, we describe some important design details to enable the full understanding of our case study.

---

1 Mobile Information Device Profile

5.1 The JADE-LEAP Platform as an Integrated Distributed Environment

The case study consists of a Multi-Agent System platform that supports applications that provide services to users that request them through different devices (e.g. mobile phone, smartphone and *PDAs*) using a computer connected to the *JADE-LEAP* platform. In this scenario, we use the *JADE-LEAP/PJAVA* or *JADE-LEAP/MIDP* platform to implement the mobile device application. The *JADE-LEAP* composition has provided a suitable communication protocol for small devices. The agents that run in mobile devices have to be able to communicate wirelessly with service-providing agents running in fixed computers. Thus, *JADE-LEAP* works at a very high communication level, establishing *TCP/IP* connections between containers, without caring about the physical means by which these connections are actually performed. The *JADE-LEAP/PJAVA* has two execution modes.

Figure 2 illustrates both execution modes.

If it is not possible to run the *JADE-LEAP* container in the device because of its limitations, the device can run the platform in "split execution mode." This mode allows the device to share resources with another computer on the platform. Thus, a "heavy" part of the container (*JADE-LEAP* BackEnd) runs on a computer that is integrated in the *JADE-LEAP* platform and a "light" part (*JADE-LEAP* FrontEnd) runs in the limited device without overloading it. The *JADE-LEAP/MIDP* platform has only the split execution mode, and is intended for strongly limited devices. This process is transparent to the user. But if the device is able to run the container using its own resources, the device runs it in stand-alone execution mode. In this mode the container is located inside the device.

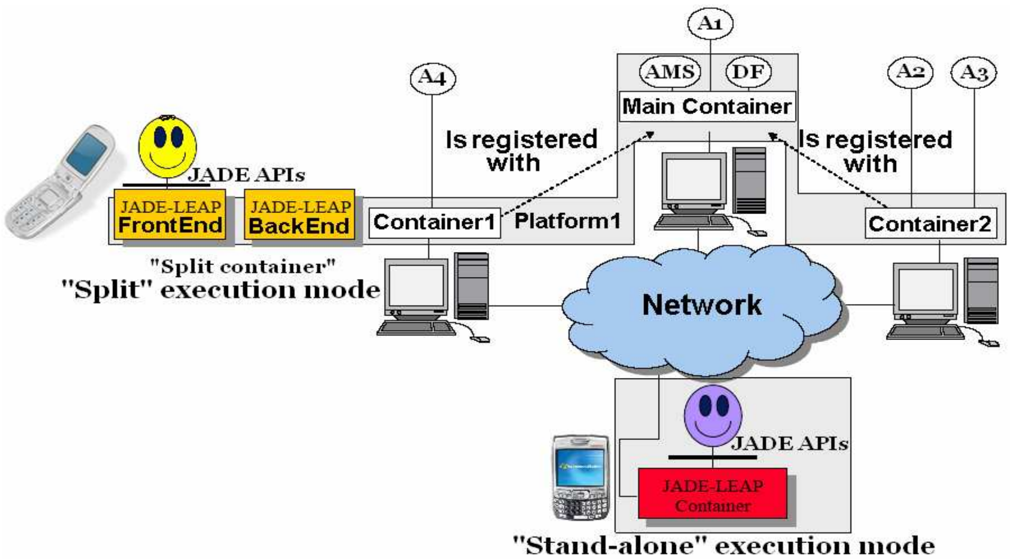

Figure 2: JADE-LEAP Execution Modes

5.2 The JADEX Platform as an Intelligent Distributed Environment

The *JADEX* Project is conducted by the Distributed Systems and Information Systems Group at the Hamburg University [Distributed Systems and Information Systems Group 2007] [Pokahr et al. 2007] [Pokahr and Braubach 2007 [d]]. It consists of a framework that includes various tools for runtime and debugging activities as well as for the development and documentation of the solutions found. The *JADEX* main proposal consists of extending the *JADE* Platform using the reasoning concept and the Belief Desire Intention Model. Figure 3 presents the *JADEX* Abstract Architecture.

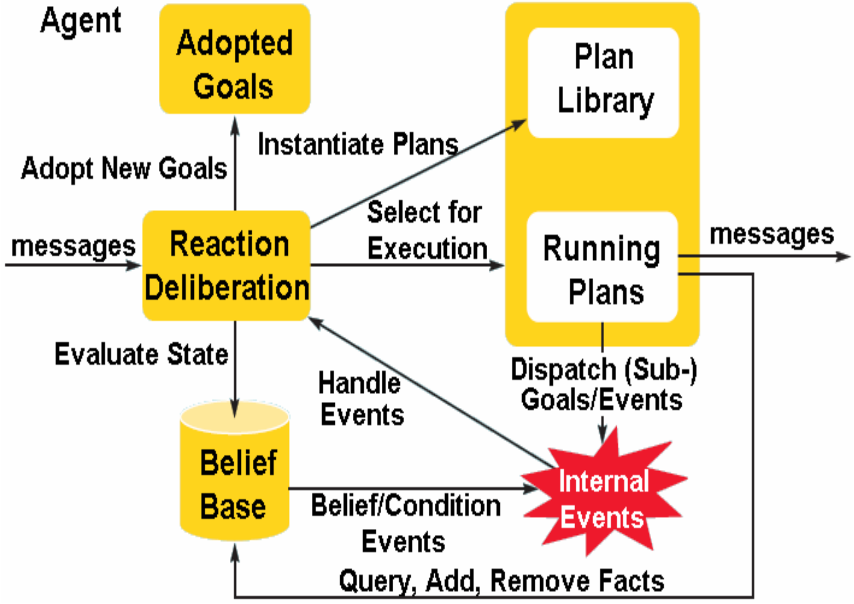

Figure 3: JADEX Abstract Architecture[2]

JADEX is an implementation of a reactive and deliberative agent architecture for representing mental states following the *BDI* model. Messages deliberate reactions that instantiate plans based on Plan Library. The Plan is selected and executed. Frequently, new goals are adopted and the Belief Base is consulted. When the plan is running, it dispatches goals, sub-goals and events, as internal events or message events. More details about the *JADEX* Execution Mode are illustrated in Figure 4.

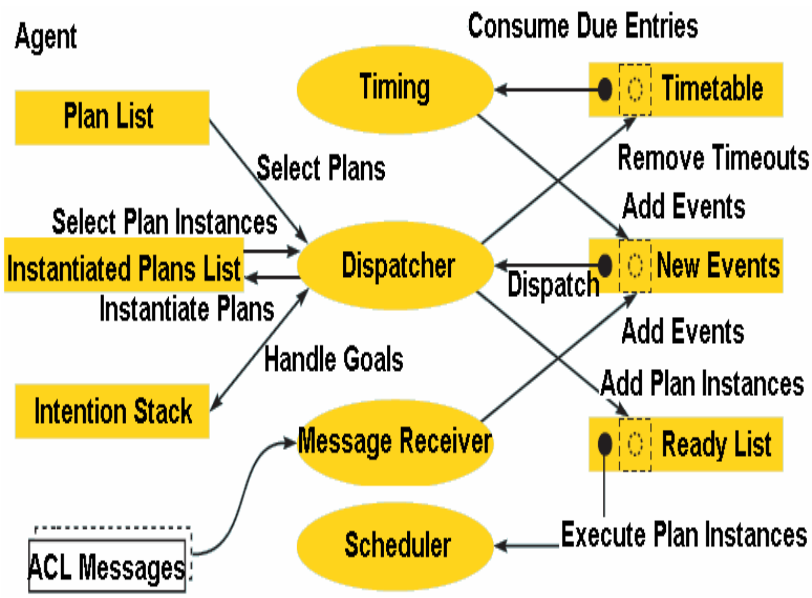

Figure 4: JADEX Execution Mode[3]

2 Part of this figure is based on [Braubach et al. 2003].

The *JADEX* functionalities are implemented in behaviors that run inside the *JADEX* agent and operate concurrently on the internal agent data-structures. These behaviors include the scheduler, the dispatcher, the message receiver, and the timing and they are associated with timetable, new events and ready lists and are responsible for the addition, removal and instantiation operations. More information about this process is presented in section 5.4. We use the *JADEX* abstract architecture and *BDI* Model to organize, to perform the plans, goals, beliefs and other agent capabilities, and to provide part of the agents reasoning. The reasoning is completed by using Fuzzy Logic as presented in section 5.7.

5.3 The Application Layer Structure

Basically, using a bottom up description and considering Figure 5, the first layer is the database used to maintain the profile information; the second is the persistence layer used to provide the database and application integration; the third layer is the *JADE-LEAP* layer that represents the *JADE-LEAP* Platform; the fourth is an adapter used to allow the integration of the *JADEX* and *JADE-LEAP* platforms; the fifth is the *JADEX* layer that represents the *JADEX* Platform based in the Belief Desire Intention Model; the sixth is both the Multi-Agent System Layer and Interface Layer that run in the access device.

It is important to observe that the limited device runs over the *JADE-LEAP* Platform as explained in section 5.1. The agent *LEAP* is very "light" and does not need to be concerned about other layers. These worries are delegated to the Multi-Agent System.

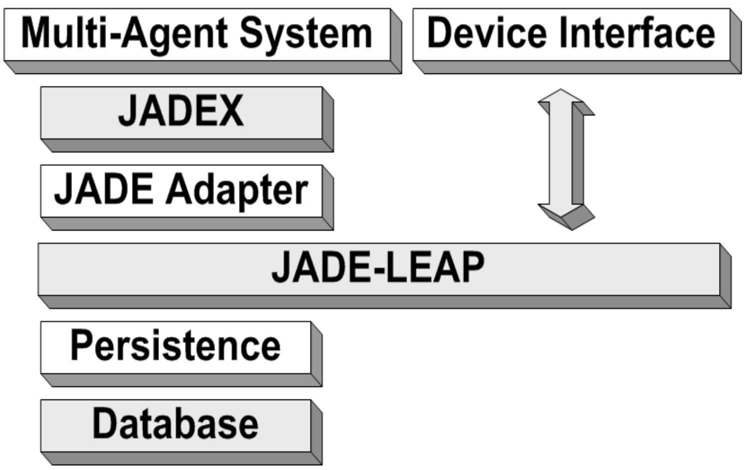

Figure 5: Application Layer Structure

5.4 The Application Modules' Architecture

The application abstract architecture is organized in modules. There are four key modules: Main Module, *JADE-LEAP* Module, Adaptation Module and Profiles Module. They always work together to provide all the application functionalities. Short descriptions of these modules are presented below. Figure 6 illustrates how they are organized and distributed in the ubiquitous context.

3 Part of this figure is based on [Braubach et al. 2003].

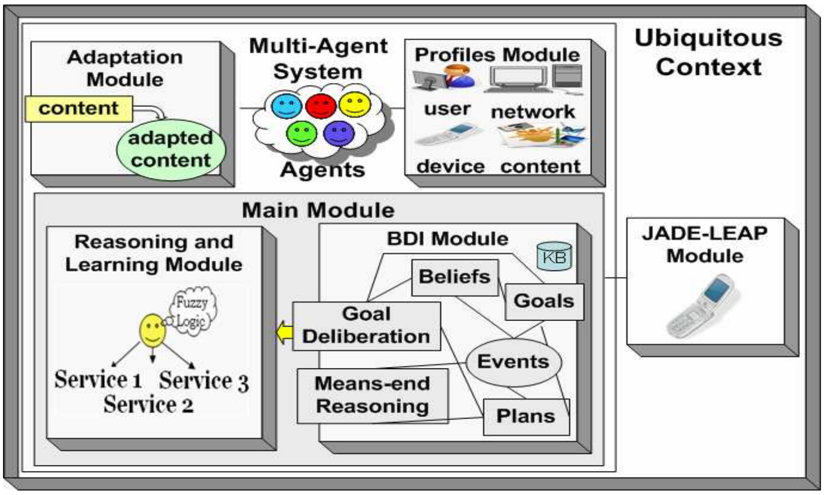

Figure 6: Application Modules' Architecture

5.4.1 The Main Module

The core module is subdivided into: (i) the Reasoning and Learning Module; and (ii) the *BDI* Module. The first implements the fuzzy logic, reasoning and learning techniques. In this module the main agent is the Analyst. The last module contains the *BDI* (Belief Desire Intention) Model that considers beliefs, desires and intentions as mental activities to simulate human behavior. In this module we have goals, beliefs, plans, events, goal deliberations, means-end reasoning and other *JADEX* agent's capabilities. This process is illustrated in Figure 7. The *JADEX* agents are basically formed by a practical reasoning interpreter and a capability. The interpreter involves goal deliberation to choose goals and means-end reasoning to select and execute the plans. The deliberation is activated by handling deliberation situations dispatched by the agent beliefs. The reasoning is activated by handling events such as incoming messages, internal events and goals. A parallel between the *JADEX* and the *JADE* Platforms establishes that the *JADEX* agent beliefs, goals and plans are determined by the programmer and represent the *JADE* agent behavior.

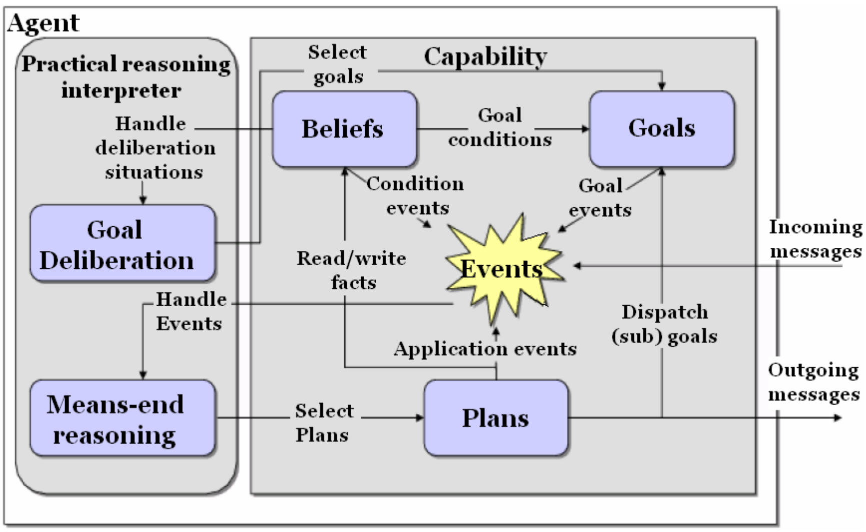

Figure 7: *BDI* Model

5.4.2 The *JADE-LEAP* Module

This module considers the mobile devices that are limited in terms of internal memory (e.g. *MIDP* mobile phones). We have used the Sun Wireless Toolkit that emulates a mobile device to implement this module. In this module we have the *JADE-LEAP* platform running inside the device and, when the user wants to request a service, the device connects via wireless network to a computer connected to the platform. At this point, the device is integrated to the platform and, for example, the agents can be registered or search other agents in the Directory Facilitator (*DF*) or even a service request can be provided. In other words, the device is part of the platform and can use all the resources it offers.

5.4.3 The Adaptation Module

This module deals with content adaptation. Sometimes it is necessary to adapt the content to satisfy the user preferences, the device capabilities, the content information and the network specifications. This module depends on the Profiles Module. At present, we deal with adaptation at the same application server, but we plan to extend it to use a dedicated server to improve the content adaptation. It is a good practice to avoid overloading the application server.

5.4.4 The Profiles Module

This module gathers all the profiles that are presented in section 4.4, such as user and device profiles. Basically, the persistence module provides the information retrieval, storage and updating.

All the described modules are integrated into the Multi-Agent System. The agents collaborate with each other to perform their tasks and to satisfy the users. Thus, for example, if an agent cannot solve a problem by using its own knowledge base, then other agents can assist it with their knowledge and the chances of solving the problem are considerably enhanced.

5.5 The Application Agents

The application involves different agents to provide the application functionalities and the services requested by the user. Some of them are listed below:

- **User Agent:** is a *JADE-LEAP* agent that runs in the user device and communicates with Core Agent to request a service. It is a very "light" agent that delegates the search, the analyses and the adaptation of the service to the Core Agent.
- **Core Agent:** is the most important agent in our Multi-Agent System design. It centralizes different user's requests, maintains the knowledge base using learning techniques and creates an Analyst Agent when necessary. The Core Agent behaves like a cache for service requests. This agent conducts the search, the analyses and the adaptation of the service to avoid the User Agent overload. It verifies its knowledge base to know if it is possible to answer the service request with its own resources. If it is not possible, then it creates an Analyst Agent to solve the problem.

- **Analyst Agent:** is created by the Core Agent to help it with the service search. Analyst Agent represents the user in the Multi-Agent System if the Core Agent delegates it. It uses reasoning and fuzzy logic to decide which service is the best to be sent to the user. Thus, this agent collaborates with the Core Agent to satisfy the user.
- **Music Store Agent:** represents an intermediary that contains content information, such as a *URL*, a price, security and number of services. One or more Music Store Agents can interact with Analyst Agent to exchange proposals and to inform the service specifications.

We have established several protocols based on *FIPA* performatives and the *ACL* Language [Ametller et al. 2003] to control the communication among the agents.

5.6 The Application Modeling

Modeling is a central Software Engineering activity and different approaches have been used to address it. As our application uses the *BDI* Model, the actors and agents are based in the Goal-Oriented Paradigm. Thus, we chose the iStar Framework and the *OME* Modeling Tool [Yu 2001] to model the ubiquitous application functional and non-functional requirements [Chung et al. 2000]. iStar representation facilitates decisions and the mapping of the decisions for future consultations and analyses. It is a way to maintain the traceability between requirements and code. This practice aims at Evolutionary Maintenance (e.g. change management and change impact analysis) and allows the reuse of the business model and the software.

5.7 Reasoning with Fuzzy Logic

To provide reasoning capability for the agents, we combine the use of the *BDI* Model, presented in sections 5.2 and 5.4, and Fuzzy Logic. The formal concept of a fuzzy set was introduced by Lofti A. Zadeh in 1965 [Zadeh 1965]. In the fuzzy set theory, each set element has a pertinence degree and it is appropriated to be used with uncertainty. Thus, we consider that fuzzy logic was the best way to classify and to analyze service proposals, based on fuzzy information that is exchanged between the agents. Our Analyst Agent, for example, analyzes the Music Store Agents proposals using several fuzzy conditional rules and reasoning techniques to decide which service proposal is considered the best. To combine all the rules and to perform the analyses, we use the *JADEX* Belief Desire Model and a Fuzzy Logic Library [Bigus and Bigus 2001].

Figure 8 illustrates the some fuzzy variables, defined by using at least five fuzzy sets. The variables are organized in: (i) **Fuzzy Input Variables** such as Price with 6 fuzzy sets; and (ii) **Fuzzy Output Variable** as Quality with 7 fuzzy sets.

To exemplify, the fuzzy set Price is classified as "free", "cheap", "medium", "high", "expensive" and "abusive" and the fuzzy set Quality is classified in "terrible", "low", "bad", "medium", "good", "high" and "excellent." We have determined the quality values based on the variable combinations. Some rules are formed using only one variable and combining the variables forms others. The Analyst Agent considers the price, the security and the number of services that are informed. Thus, using conditional rules, the Analyst Agent establishes the proposal quality values and decides which proposal is the best to satisfy the user request.

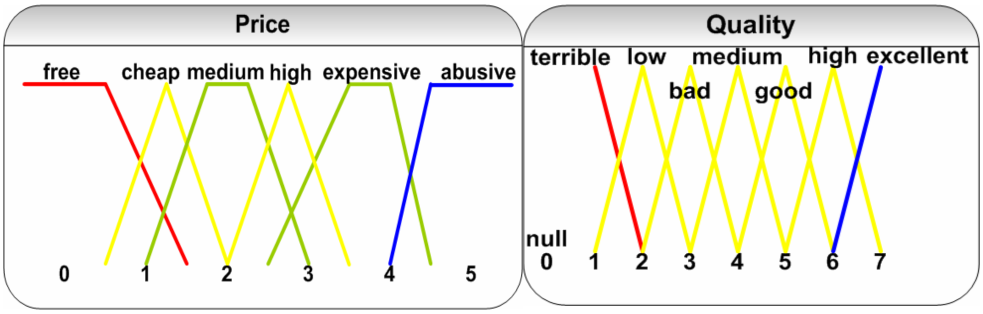

Figure 8: Fuzzy Logic Sets

5.8 The Application Execution and the First Results

First of all, we have to consider the application that was developed. Figure 9 illustrates the application execution. It is running in two notebooks and one mobile device. One notebook is running the "Main-Container" of the platform. The other notebook represents another computer that belongs to the platform and it is running the "Container-1." We can have several desktops or notebooks integrated in this ubiquitous environment. Now, imagining that the user mobile device is limited by its internal memory capacity, then it has to use the split execution mode (already explained in section 5.1). This mode allows the device to share resources with another computer that has more powerful capabilities.

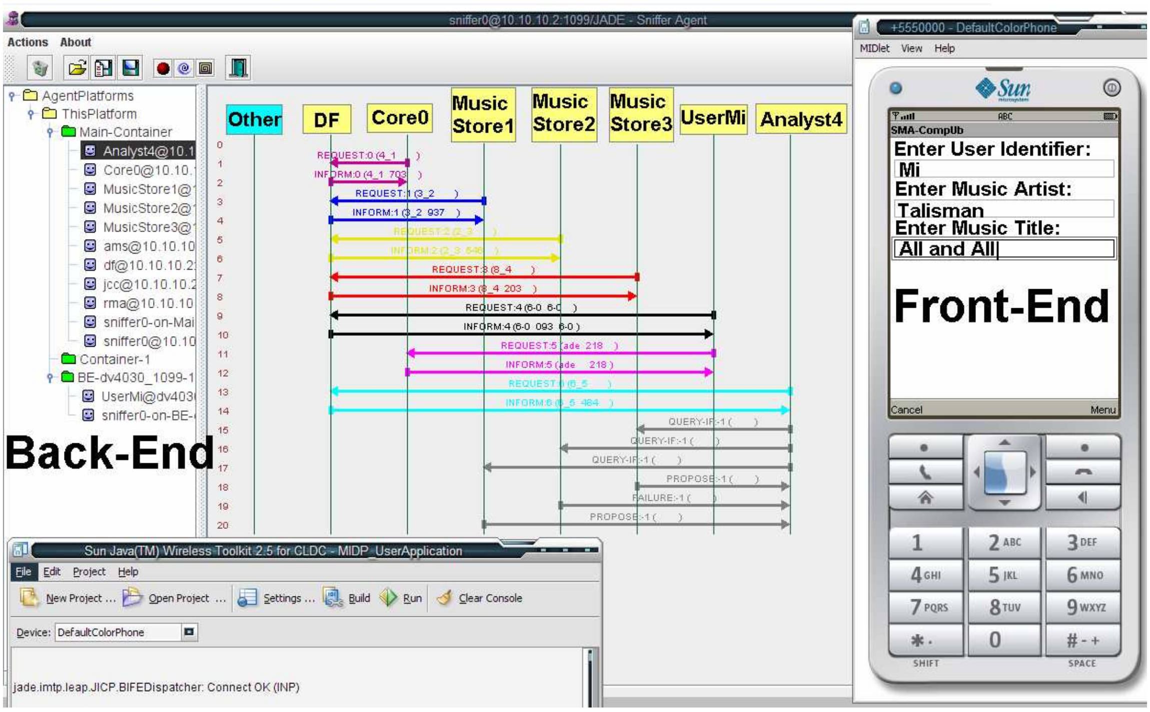

Figure 9: Application Execution

When the device connects with the notebook that is running "Container-1," through a wireless network, it requests that a "heavy" part of the container, called "Back-End (BE)," be maintained in that notebook. The other part, called "Front-End (FE)," is lighter than the first and runs in the mobile device. It is possible to observe that the agents are exchanging information. The Core Agent and the Music Store Agents request their registers at Yellow Pages from the *DF* Agent.

The User Agent that runs in the mobile phone requests a Core Agent Identifier from *DF* Agent. This latter agent informs the User Agent about the Core Agent. Then the User Agent requests the service from Core Agent. Now, the Core Agent has full responsibility to find the service, because the User Agent delegated it. Some other agent design details are showed in Figure 10 that is organized in beliefs, goals, plans, modules and main functionalities for each application agent.

Agents	User Agent	Core Agent	Analyst Agent	Music Store Agents
Beliefs <<name/class>>	user/User core /Core url/String	active-users/User service-cache/ ServiceQuality musics/Music	music-store/AgentIdentifier service-reports/ServiceReport reasoning-results/ServiceQuality fuzzy-rule-base FuzzyRuleBase	reports/ServiceReport musics/Music
Goals	df_search find-core obtain-agent	ams-create-agent df-register df-deregister user-be-attended inform-user	df-search find-music-store setup-reasoning-engine avail-all-reports aswer-core	df_register df-deregister
Plans	InformCorePlan.java SearchCorePlan.java	GetServiceFromCachePlan.java ReceiveUserRequestPlan.java SendServiceToUserPlan.java UserBeAttendedMetaPlan.java	AnalystServiceReportPlan.java CreateFuzzyRuleBasePlan.java ReceiveReportPlan.java SendBestReportPlan.java	SearchMusicPlan.java SetupPlan.java
Modules	JADE-LEAP	BDI and Reasoning	Reasoning	BDI and Reasoning
Main Functionalities	Represents the User Agent and Interacts with Core Agent	Interacts with User Agent to satisfy him and manufactures Analyst Agent	Analyzes the services proposes using reasoning, determines the best one of them and interacts with MusicStore Agents	Present the information about the music content and interact with Analyst Agent.

Figure 10: Some Agent Design Details

We observe that an Analyst Agent is part of the process, thus the Core Agent knowledge base could not answer the user request and an Analyst Agent was created to do this. The Analyst Agent searches other agents in Yellow Pages. The *DF* Agent informs it about the registered Music Store Agents. Thus, it consults them and receives two service proposals from Music Store 1 and Music Store 3 and one failure from Music Store 2. Based in these two proposals, the Analyst decides which proposal is the best one using Fuzzy Logic as the reasoning technique. It chooses the proposal informed by the Music Store 3 and sends it to Core Agent, which verifies the profiles specifications. If the Core Agent identifies the content adaptation needed, it adapts it and informs the User Agent that the service was found. Finally, the user receives the service at his/her device. Note that all activities involved in the process, including service search, proposals analyses, content adaptation, knowledge base update, and service delivery, are the responsibility of the Multi-Agent System that runs in a powerful computer. The User Agent does not perform anything, only delegates it to the Core Agent.

Analyzing the application construction, we can see that the *JADE-LEAP* platforms are integrated according to section 5.1. This application allows the user to request and to receive the desired service through his/her mobile phone. This ubiquitous environment illustrates that is possible to offer services anywhere and any time using only computer machines distributed in the network and integrated in the *JADE-LEAP* platform (see Figure 11). The Multi-Agent System and *JADEX BDI* Model really seem to be very convenient for dealing with Ubiquitous Computing concerns such as device memory limitations, content adaptation, different devices capabilities and inherent user mobility.

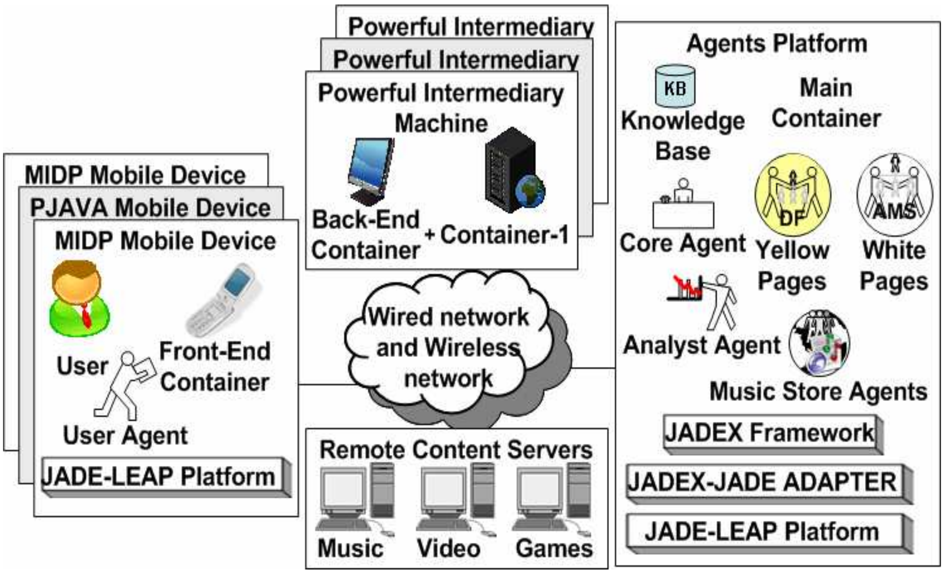

Figure 11: Ubiquitous Environment and the Proposed Solution

6 Conclusions and Future Work

Finally, we conclude that some of our main research challenges and hypotheses have been met and confirmed. Others remain to be analyzed in our future work. We are convinced that the case study indicates that we are moving in the right direction.

Agent properties, especially mobility and autonomy, are very interesting for the domain of ubiquitous applications. Moreover, the agent collaboration and the use of reasoning, learning and the goal-oriented planning are very adequate to solve one of the main Ubiquitous Computing concerns. This concern consists of providing services to clients that are in a distributed world and that have only limited mobile devices to access them.

In addition, the Belief Desire Intention Model with goals, plans, beliefs, plans deliberations, events and other *JADEX* agent capabilities is useful to allow an appropriate goal-oriented modeling. With this modeling approach it is easy to maintain the ubiquitous requirements of traceability, including functional and non-functional requirements. Therefore, Evolutionary Maintenance can be realized based on the previous deployment decisions only by analyzing the goal-oriented model.

There are several theories, principles, technologies, concepts, features and assumptions that we still need to examine during the course of our research. They include: (i) to investigate additional technologies that may be interesting to the ubiquitous context; (ii) to map other traditional Software Engineering solutions to Ubiquitous Computing; (iii) to study new Software Engineering approaches specific to Ubiquitous Computing; (iv) to compare the multi-agent-based results with other paradigms, such as Object and Aspect Orientation, and (v) to use the experimental environment developed and presented in this paper to test and evaluate new applications and emerging technologies.

References

Agotnes, T.; Hoek, W. van der; Wooldridge, M. "Reasoning about Judgment and Preference Aggregation." Proceedings of Sixth International Conference on Autonomous and Multiagent Systems, AAMAS'07, pp.554-561, Honolulu, Hawaii, USA, May 2007.

Ametller, J.; Robles, S.; Borrell, J. "Agent Migration over FIPA ACL Messages." Mobile Agents for Telecommunication Applications. 5th International Workshop, MATA, vol.2881, pp.210-219, Marrakech, Morocco, October 2003.

Assad, M.; Carmichael, D. J.; Kay, J.; Kummerfeld, B. "PersonisAD: Distributed, active, scrutable model framework for context-aware services." In Proceedings of PERVASIVE 07, 5th International Conference on Pervasive Computing, vol.4480 of Lecture Notes in Computer Science, pp.55-72. Springer-Verlag, Toronto, Ontario, Canada, 2007.

Bellavista, P.; Corradi, A.; Stefanelli, C. "Mobile Agent Middleware for Mobile Computing." University of Bologna and University of Ferrara, IEEE, pp.73-81, March 2001.

Bellifemine, F.; Caire, G.; Greenwood, D. "Developing Multi-Agent Systems with JADE." John Wiley & Sons, pages 286, ISBN 0470057475, April 2007.

Bigus, J.; Bigus, J. "Constructing Intelligent Agents Using Java." Second Edition. Ed. John Wiley & Sons Incorporated, 2 Edition, pages 432, ISBN 978-0471396017, March 2001.

Braubach, L.; Lamersdorf, W.; Pokahr, A. "Jadex: Implementing a BDI Infrastructure for JADE Agents." Distributed Systems and Information Systems, University of Hamburg, vol.3, n.3, pp.76-85, September 2003.

Brumitt, B. L.; Meyers, B.; Krumm, J.; Kern, A.; Shafer, S. "EasyLiving: Technologies for intelligent environments." Proc. International Symposium on Handheld and Ubiquitous Computing (HUC'00), pp.12-27, September 2000.

Caire, G. "LEAP User Guide." Copyright (C) TILAB, LEAP3.1, December 2003.

Carrillo-Ramos, A.; Gensel, J.; Villanova-Oliver, M.; Martin, H. "PUMAS: a Framework based on Ubiquitous Agents for Accessing Web Information Systems through Mobile Devices." Laboratoire LSR – IMAG, ACM Symposium on Applied Computing, pp.1003-1008, 2005.

CHAI - Computer Human Adapted Interaction Research Group – Home Page, University of Sydney, Australia, 2007. Accessed at: http://chai.it.usyd.edu.au/index.cgi?Context=Home

Chen, E.; Sabaz, D.; Gruver, W. A. "Wireless Distributed Systems with JADE." IEEE International Conference on Systems, Man and Cybernetics, pp.989-993, 2004.

Chung, L. K.; Nixon, B. A.; Yu, E.; Mylopoulos, J. "Non-Functional Requirements in Software Engineering." Kluwer Academic Publishers, vol.5, 472 pages, ISBN 978-0-7923-8666-7, 2000.

Distributed Systems and Information Systems Group – Home Page, University of Hamburg, Germany, 2007. Accessed at: http://vsis-www.informatik.uni-hamburg.de

Embedded Systems and Pervasive Computing Lab Group – Home Page, Federal University of Campina Grande, Brazil, 2007. Accessed at: http://embedded.ufcg.edu.br/indexen.html

Garlan, D.; Siewiorek, D.; Smailagic, A.; Steenkiste, P. "Project Aura: Towards distraction-free pervasive computing." IEEE Pervasive Computing, vol.1, pp.22–31, 2002.

Gay, V.; Leijdekkers, P. "Around the Clock Personalized Health Monitoring Using Smart Phones." 4th International Conference on Smart Homes and Health Telematics ICOST'06, Belfast, Ireland, June 2006.

Helin, H.; Pelt, T. van; Schumacher, M.; Syreeni, S. "Efficient Networking for Pervasive eHealth Applications." CASCOM Research Project, First European Conference on EHealth (ECEH'06), Fribourg, Switzerland, October 2006.

Kapitza, R.; Schmidt, H.; Soldner, G.; Hauck, F. J. "A framework for adaptive mobile objects in heterogeneous environments." In OTM06, vol. 4276 of LNCS, pp.1739–1756, 2006.

Lagos, M. E.; Alarcón, R.; Nussbaum, M.; Capponi, F. "Interaction-Based Design for Mobile Collaborative-Learning Software." Pontificia Universidad Católica de Chile, Software Development Feature, IEEE SOFTWARE, Published by the IEEE Computer Society, pp.80-89, July/August 2007.

Leijdekkers, P.; Gay, V. "Personal Heart Monitoring and Rehabilitation System using Smart Phones." IEEE International conference on mobile business, Copenhagen, Denmark, June 2006.

Lemaître, C.; Reyes C. A.; González, J. A. "Advances in Artificial Intelligence." IBERAMIA, LNAI 3315, 9th Ibero-American Conference on AI, Springer Publisher, 987 pages, ISBN 978-3-540-23806-5, 1st Edition, Puebla, México, November 2004.

Liu, Y. "Enabling adaptation of J2EE applications using components, web services and aspects." In Proceedings of the 5th Workshop on Adaptive and Reflective Middleware (ARM'06) (Melbourne, Australia, December), vol.190, ACM, New York, N.Y., 2006.

Lobato, C., Garcia, A., Lucena, C., and Romanovsky, A. "A modular implementation framework for code mobility." In Proceedings of the 3rd international Conference on Mobile Technology, Applications &Amp; Systems (Bangkok, Thailand), Mobility '06, vol.270, ACM, ISBN 1-59593-519-3, New York, NY, 2006.

Mobile Ubiquitous Services & Technologies Group – Home Page, ARN Advanced Research in Networking, Department of Computer Systems, Faculty of Information Technology, University of Technology, Sydney, Australia, 2007. Accessed at: http://www-staff.it.uts.edu.au/~peterl/mobilelab/index.php

Moraïtis, P.; Petraki. E.; Spanoudakis, N. I. "Engineering JADE Agents with the Gaia Methodology". Department of Computer Science, University of Cyprus, Springer Berlin / Heidelberg, pp.77-91, 2003

NICTA Group – Home Page, Context Aware Mobility Project, Australia, 2007. Accessed at: http://nicta.com.au/research/projects/context_aware_mobility_project/camp_detail

Nokia Group. "MIDP: Wireless Messaging API 2.0 Developer's Guide." Forum Nokia, Version 1.0, November 2005.

Pokahr, A. and Braubach. L. "JADEX Tool Guide." Distributed Systems Group, University of Hamburg, Germany, Release 0.96 - Last Update: June 2007. Accessed at: http://vsis-www.informatik.uni-hamburg.de/projects/jadex

Pokahr, A. and Braubach. L. "JADEX Tutorial." Distributed Systems Group, University of Hamburg, Germany, Release 0.96 - Last Update: June 2007. Accessed at: http://vsis-www.informatik.uni-hamburg.de/projects/jadex

Pokahr, A. and Braubach. L. "JADEX User Guide." Distributed Systems Group, University of Hamburg, Germany, Release 0.96 - Last Update: June 2007. Accessed at: http://vsis-www.informatik.uni-hamburg.de/projects/jadex

Pokahr, A.; Braubach, L. "An Architecture and Framework for Agent-Based Web Applications." In: 5th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS'07), 2007.

Pokahr, A.; Braubach, L.; Lamersdorf, W. "Jadex: A BDI Reasoning Engine". Programming Multi-Agent Systems, Kluwer Academic Publishers, pp.149-174, 2005.

Pokahr, A.; Braubach, L.; Walczak, A.; Lamersdorf, W. "Jadex - Engineering Goal-Oriented Agents." In: F. Bellifemine and G. Caire and D. Greenwood (Hrsg.): Developing Multi-Agent Systems with JADE, Wiley & Sons, pp.254-258, 2007.

Román, M.; Hess, C. K.; Cerqueira, R.; Ranganat, A.; Campbell, R. H.; Nahrstedt, K.; Gaia, K. "A middleware infrastructure to enable active spaces." IEEE Pervasive Computing, vol.1, pp.74-82, 2002.

Ruano, A. E. "Intelligent Control Systems using Computational Intelligence Techniques." IEE Control Series 70. Published by IET, 667 pages, ISBN 0863414893, 2005.

Satoh, I†. "Mobile Applications in Ubiquitous Computing Environments." Copyright The Institute of Electronics, Information and Communication Engineers, SATOH: Mobile Applications in Ubiquitous Computing Environments, IEICE TRANS. COMMUN., vol.E88–B, n.3, pp.1026-1033, March 2005.

Tandler, P. "Software infrastructure for ubiquitous computing environments: Supporting synchronous collaboration with heterogeneous devices." Proc. UbiComp'2001, LNCS, vol.2201, pp.96–115, Springer, 2001.

Wegdam, M; Bemmel1, J. van; Lagerberg, K.; Leijdekkers, P. "An Architecture for User Location in Heterogeneous Mobile Networks." In: Proceedings of the 7th IEEE International Conference on High-Speed Networks and Multimedia Communication (HSNMC'04), Springer Berlin/Heidelberg, vol.3079, pp.479-491, Toulouse, France, July 2004.

Yu, E. "Agent-Oriented Modeling: Software Versus the World." In Agent-Oriented Software Engineering AOSE-01 Workshop Proceedings, vol.2222, pp.206-225, Montreal, Canada, May 2001.

Zadeh, L. A. "Fuzzy sets. Information and Control." vol.8, pp.338-352, 1965. Accessed at: http://www.cs.berkeley.edu/~zadeh/ (Author's Home Page, 2007)